

Title of the Invention:

筋電パターン識別方法及び装置

(MYOELECTRIC-PATTERN CLASSIFICATION METHOD AND APPARATUS)

BACK GROUND OF THE INVENTION

Field of the Invention:

【0001】

本発明は、筋肉の活動電位（筋電パターン）をインターフェースとする方法における筋電パターン識別方法及び装置に関する。

Description of the Prior Art:

【0002】

図2は、従来技術の筋電パターンから特徴量を抽出し、ビット列に符号化し、符号化されたビットパターン識別を行って、対象を操作する装置を説明するための図である。図中、①は筋電パターン、②は表面電極群、③は増幅、平滑化装置、④は特徴パターン抽出装置、⑤は符号化装置（バイナリ符号・グレイ符号）、⑥はパターン識別器、⑦はモータ、ロボット、福祉機器、リハビリテーション機器、筋電義手、ゲームなどの制御対象物をそれぞれ示している。

【0003】

図示したように、複数の筋肉の協調活動から発生する活動電位である筋電パターン①は、皮膚表面上の一つ、または複数個の電極群②で測定される。ここで測定されるのは、複数の筋肉から発生した活動電位の和である。次に、得られた電位の和は、増幅、平滑化装置③によって増幅、平滑化を施される。特徴パターン抽出装置④では、増幅、平滑化した信号より特徴パターンを抽出する。符号化装置⑤では、得られた特徴パターンを、バイナリ符号やグレイ符号を用いて、ビット列に符号化する。パターン識別器⑥では、符号化されたパターン識別を行い、制御対象物⑦の制御信号を生成する。

【0004】

このような従来技術は、符号化に表1に示すようなバイナリ符号や、グレイ符号を用いるために、パターン識別器の設計に時間がかかる場合や、複雑なパターン識別器が必要な場合がある。このために、小型化、低価格化が妨げられ、筋電パターンをインターフェースとする装置の普及が妨げられてきた。

## 【0005】

【表1】

特徴量	グレイ符号 $X_1 X_2 X_3 X_4$	バイナリ符号 $X_1 X_2 X_3 X_4$
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111

## 【0006】

また、バイナリ符号、グレイ符号の場合は、自発的でない筋張力変化による筋電パターンの変化によって、正確なパターン識別が困難になる場合があり、このために、適用可能な範囲が限られてきた。

## 【0007】

本発明の目的は、上述の問題点を解決して、単純な演算処理装置や、表引き装置によって実現が可能で、小型化、低価格の筋電パターン識別方法及び装置を提供することにある。

## 【0008】

この発明の他の目的は、識別精度を改善し、筋電をインターフェースとする方法、装置の普及を促進する筋電パターン識別方法及び装置を提供することにある。

### SUMMARY OF THE INVENTION

## 【0009】

本発明の筋電パターン識別方法は、複数の筋肉の協調活動から発生する活動電位である筋電パターンを皮膚表面上で表面電極群により測定し、測定された

筋電パターンから特徴パターンを抽出し、抽出された特徴パターンを冗長符号によってビットパターンに符号化し、符号化されたビットパターンのパターン識別を行って出力制御信号を生成することから成る。

【0010】

また、本発明の筋電パターン識別装置は、複数の筋肉の協調活動から発生する活動電位である筋電パターンを皮膚表面上で測定する表面電極群と、測定された筋電パターンから特徴パターンを抽出する特徴パターン抽出装置と、抽出された特徴パターンを冗長符号によってビットパターンに符号化する符号化装置と、符号化されたビットパターンの識別を行って出力制御信号を生成するパターン識別器とから成る。

【0011】

上述の如く、本発明では特徴パターンを冗長符号によってビットパターンに符号化することにより、冗長符号は、任意の連続的な値に対しては、常に1ビットだけ異なり、差が2の場合は2ビット、差が3の場合は3ビット異なるため、識別回路を簡単な回路で実現できる。

【0012】

この発明の他の目的、その他の特徴は添付の図面に基づく以下の詳しい説明で明らかにする。

BRIEF EXPLANATION OF THE DRAWING

【0013】

図1は、本発明に依る筋電パターン識別装置の構成説明図である。

【0014】

図2は、従来の筋電パターン識別装置の構成説明図である。

【0015】

図3は、筋電を測定するための装置の概略構成図である。

【0016】

図4 (a) は、2秒間の脱力後、3秒間筋収縮させた状態を示すグラフである。

【0017】

図4 (b) は、図4 (a) の状態で手首を屈曲した時の筋電信号の一例を示

すグラフである。

【0018】

図4(c)は、図4(b)の筋電信号を平滑化処理した筋電信号の一例を示すグラフである。

【0019】

図5は、特徴パターンの抽出方法を説明するための図である。

【0020】

図6は、筋電の特徴パターンの抽出方法パターンの分布図である。

【0021】

図7は、Programmable Logic Array(PLA)の構造を説明するための図である。

DESCRIPTION OF THE PREFERRED EMBODIMENT

【0022】

図1は本発明の筋電パターンから特徴量を抽出して、制御対象物を操作する筋電パターン識別装置の一実施例を示し、①は筋電パターン、②は表面電極群、③は増幅、平滑化装置、④は特徴パターン抽出装置、⑧は冗長符号による符号化装置、⑥はパターン識別器、⑦はモータ、ロボット、筋電義手などの制御対象物をそれぞれ示している。

【0023】

図示したように、複数の筋肉の協調活動から発生する活動電位である筋電パターン①は、皮膚表面上の一つ、または複数個の電極群②で測定される。ここで測定されるのは、複数の筋肉から発生した活動電位の和である。次に、それらは、増幅、平滑化装置③によって増幅され、必要に応じて平滑化を施される。

【0024】

特徴パターン抽出装置④では、特徴パターンを抽出する。抽出された特徴パターンは符号化装置⑧において、表2に示す冗長符号によってビットパターンに符号化される。パターン識別器⑥では、符号化されたビットパターンのパターンの識別を行い、制御対象物⑦の制御信号を生成する。パターン識別器としては、論理回路などの論理値フィルタが用いられる。

【0025】

【表2】

特徴量	冗長符号
	$X_1X_2X_3X_4$
0	0000
1	0001
2	0011
3	0111
4	1111
5	1110
6	1100
7	1000

#### 【0026】

次に、本発明に係る筋電パターン識別を筋電義手の動作決定に適応して説明する。

#### 【0027】

先ず、筋電の測定を図3により説明する。筋電は数  $\mu$ V～数mV程度の微弱な信号であるため、その測定においては、一般的に、2個の電極で測定した信号を差動増幅することによって、電源やその他の機器から発生するノイズの影響を小さくする必要がある。そこで、図示の実施例では8個（4組）の電極12を筋電義手の前腕11周りに張り付けて筋電を測定する。13は参照電極である。

#### 【0028】

4組の電極12で測定した微弱な筋電信号は、Sensor Box 14を介して増幅器15へ送られて差動増幅し、次いでA/D Converter 16へ送られ、A/D Converter 16では増幅された筋電信号をそれぞれ1000Hzで隔離化し、12ビットのデジタル信号に変換してSCSIインターフェースで接続しているノート型パソコン（ノートPC）17へ取り込む。図中、18は電源、19はアースである。また、この測定においては、商用電源から発生するノイズを取り除くために、商用電源周波数（50Hz）を遮断するアナログフィルタを用いた。

### 【0029】

以下の説明では、時刻  $t$  で測定した信号の組を、数1であらわすことにする。

### 【0030】

#### 【数1】

$$\mathbf{f}(t) = (f_1(t), f_2(t), f_3(t), f_4(t))$$

### 【0031】

ここでは、6つの動作（前腕の回内、回外、手首の屈曲、伸展、手先の開閉）を意図しながら筋肉を収縮させたときに、筋電を測定する。つまり、図4(a)に示す筋収縮（2秒間の脱力の後に3秒間筋収縮を維持する）を6つの動作毎に20回（合計120回）繰り返しているときに筋電を測定する。ただし、前腕切断者の場合は、切断前の動作イメージをもとに筋肉を収縮したときの筋電を測定する。

### 【0032】

図4(b)に、手首を屈曲した時に前腕の屈筋群上の皮膚表面で測定した筋電の例を示す。

### 【0033】

測定した筋電信号は、高周波成分を取り除くために平滑化処理を施す。一般的に、この平滑化処理にはアナログローパスフィルタを用いることが多いが、本測定では、情報を失わないためにローパスフィルタをかけずにPCに取り込んで、PC上で平滑化処理を施す。具体的には、時刻  $t$  における平滑整流値を数2に示す計算によって求める。図4(c)に平滑整流した筋電の例を示す。

### 【0034】

#### 【数2】

$$\mathbf{g}(t) = \left( \sum_{i=1}^{100} |f_1(t-i)|, \sum_{i=1}^{100} |f_2(t-i)|, \sum_{i=1}^{100} |f_3(t-i)|, \sum_{i=1}^{100} |f_4(t-i)| \right)$$

### 【0035】

義手の動作決定に用いる筋電の特徴パターンの抽出では、一般的に、(1) 筋収縮開始時に測定した筋電から抽出する方式か、(2) 筋収縮を維持した状態(以

下では定常状態と呼ぶ)で測定した筋電から抽出する方式が用いられる。(1)の方式場合、筋肉を収縮しはじめてから義手が動き出すまでの遅れが小さい利点がある。ところが、動作識別率に関しては(1)の方式よりも、(2)の方式の方が良いことが知られているので、ここでは、定常状態の筋電から特徴パターンを抽出する。

【0036】

抽出する特徴パターンとしては、定常状態の筋電の平滑整流値を用いる。この平滑化においても、一般的にはアナログローパスフィルタを用いるが、ここでは、PC上で数2で求めた値の1秒間の平均値を計算し、これを特徴パターンとする。

【0037】

この特徴パターンの抽出においては、筋電測定時の被験者の疲労を小さくするため、1回の筋収縮から10個の平均値を求める。つまり、図5に示すように、1秒毎の平均値を、動作開始1秒後から100ミリ秒づつずらして、数3に示す計算によって求める。ここで*i*は動作を開始した時間、つまり数4で求めた値が、あらかじめ設定された閾値を越えた時間とする。

【0038】

【数3】

$$ave(n) = \frac{\sum_{j=1}^{1000} g(i + 1000 + j + n * 100)}{1000}, (n = 0, 1, \dots, 9)$$

【0039】

【数4】

$$G(t) = g_1(t) + g_2(t) + g_3(t) + g_4(t)$$

【0040】

しかしながら、実際には、筋肉の収縮の仕方によっては数4の値が閾値を越えない場合もある。このため、動作毎20回の筋収縮のすべてから数3を計算できるわけではない。そこで、パターン識別器を学習させるために用いるトレーニングパターンとしては、数4の値が閾値を越えた筋収縮のうち5回から抽

出した数3の値（5 [回の筋収縮] × 10 [パターン] × 6 [種類の動作] = 300 [パターン]）を用いる。その他の5回の筋収縮から抽出した数3の値は、パターン識別器を評価するためのテストパターンとして用いる。以下の説明では、作成したトレーニングパターンを、数5で表し、作成したテストパターンを、数6で表す。

【0041】

【数5】

$$Tr_{org}(n) = (Tr_{org,1}(n), Tr_{org,2}(n), Tr_{org,3}(n), Tr_{org,4}(n)), (n = 0, 1, \dots, 299)$$

【0042】

【数6】

$$Te_{org}(n) = (Te_{org,1}(n), Te_{org,2}(n), Te_{org,3}(n), Te_{org,4}(n)), (n = 0, 1, \dots, 299)$$

【0043】

図6に、作成したトレーニングパターンの第1成分をX軸、第2成分をY軸としてプロットした例を示す。

【0044】

ここでは、さらに、日本特許出願2001-020881号に記述した対数変換を用いて筋電パターンから特徴パターンを抽出する。具体的には、以下の数7、数8を用いて対数変換を行う。

【0045】

【数7】

$$\begin{aligned} Tr_{log}(n) &= (Tr_{log,1}(n), Tr_{log,2}(n), Tr_{log,3}(n), Tr_{log,4}(n)) \\ &= (-\log(Tr_{org,1}(n)), -\log(Tr_{org,2}(n)), -\log(Tr_{org,3}(n)), -\log(Tr_{org,4}(n))) \\ &(n = 0, 1, \dots, 299) \end{aligned}$$

【0046】

【数8】

$$\begin{aligned} Te_{log}(n) &= (Te_{log,1}(n), Te_{log,2}(n), Te_{log,3}(n), Te_{log,4}(n)) \\ &= (-\log(Te_{org,1}(n)), -\log(Te_{org,2}(n)), -\log(Te_{org,3}(n)), -\log(Te_{org,4}(n))) \\ &(n = 0, 1, \dots, 299) \end{aligned}$$

### 【0047】

以下では、数5、数7で求めたトレーニングパターン、それぞれを用いてパターン識別器の訓練を行い、数6、数8で求めたテストパターンを用いて識別器の評価を行う。また、以下では、説明を簡単にするために、トレーニングパターン、テストパターンを、数9を用いてあらわす。

### 【0048】

#### 【数9】

$$\mathbf{Tr} = (Tr_1(n), Tr_2(n), Tr_3(n), Tr_4(n)), \mathbf{Te} = (Te_1(n), Te_2(n), Te_3(n), Te_4(n))$$

### 【0049】

ここでは、論理回路によって筋電パターンの識別を行うので、識別に用いるパターン識別器としては、進化型チップを用いる。この進化型チップの特徴は、従来のハードウェアは回路構成がその設計プロセスによって固定的であるのに対し、その回路構成を動的に、かつ、自律的に変更可能な能力によって、仕様の変更や動作環境の変化に対して適応できるように設計されている。

### 【0050】

進化型チップは、回路構成を何度も変更可能なハードウェア（再構成可能ハードウェア）と、その回路構成を適応的に変更する回路から構成される。この再構成可能ハードウェアの回路構成は、コンフィグレーションビット列と呼ばれるソフトウェアビット列をダウンロードすることによって、何度も変更できる。また、回路構成の適応的な書き換えには、遺伝的アルゴリズム（Genetic Algorithm；以下では GA と略す）と呼ばれる探索手法を高速に実行する回路を用いる。

### 【0051】

GA は、生物の進化に発想を得た探索手法で、複数の解の候補をそれぞれ0、1のビット列に符号化することによって、並列に最適解を探索する。つまり、これらのビット列を染色体と見なし、交叉や突然変異と呼ぶ操作を施して変化させることで、新しい解の候補を作る。そこで、あらかじめ決められた評価関数によって、各染色体（＝解の候補）がどれだけ解に近いかを評価する。この

評価結果に基づき、解から遠いものを削除し、解に近いものを残す操作を繰り返すことによって、解を探索する。

#### 【0052】

進化型チップでは、再構成可能ハードウェアの回路構成を指定するコンフィグレーションビット列を染色体と見なす。そこでGAを適用して染色体を進化させることによって、評価関数が適切に設定されていれば、最適な回路構成を指定するコンフィグレーションビット列を得ることができる。

#### 【0053】

この進化型チップは、(1) GA操作用回路、(2) 再構成可能ハードウェア、(3) 染色体を保存するメモリ(染色体用メモリ)、(4) トレーニングパターンを保存するメモリ(トレーニングパターン用メモリ)、(5) 入出力インターフェースからなる。

#### 【0054】

再構成可能ハードウェアとしては、Programmable Logic Array(以下ではPLAと略す)が用いられている。PLAは、図7に示す例のように、ANDアレイ(接続している入力信号の論理積を作る)とORアレイ(接続しているANDアレイの出力信号の論理和を作る)からなる。この図では、黒丸と白丸が、それぞれ、入力と出力間の接続を決定するスイッチを示しており(黒丸が接続を意味する)、このスイッチのON/OFFをコンフィグレーションビット列で指定することによって、任意の論理回路を構成できる。

#### 【0055】

進化型チップによるパターン識別では、論理回路によって識別を行うため、識別パターンを0、1のビット列に変換しなければならない。そこで、ここでは、筋電の特徴パターンを数10、数11に示す計算で0~15の実数に変換し、それらを整数化したものを4ビットのビット列に符号化する。

#### 【0056】

##### 【数10】

$$\begin{aligned}
 sTr(n) &= (sTr_1(n), sTr_2(n), sTr_3(n), sTr_4(n)) \\
 &= 15 \times \left( \frac{Tr_1(n) - Tr_{1_{min}}}{Tr_{1_{max}} - Tr_{1_{min}}}, \frac{Tr_2(n) - Tr_{2_{min}}}{Tr_{2_{max}} - Tr_{2_{min}}}, \frac{Tr_3(n) - Tr_{3_{min}}}{Tr_{3_{max}} - Tr_{3_{min}}}, \frac{Tr_4(n) - Tr_{4_{min}}}{Tr_{4_{max}} - Tr_{4_{min}}} \right) \\
 (n &= 0, 1, \dots, 299) \\
 (Tr_{1_{min}}, Tr_{2_{min}}, Tr_{3_{min}}, Tr_{4_{min}}) &: Tr_1(n), Tr_2(n), Tr_3(n), Tr_4(n) の最小値 \\
 (Tr_{1_{max}}, Tr_{2_{max}}, Tr_{3_{max}}, Tr_{4_{max}}) &: Tr_1(n), Tr_2(n), Tr_3(n), Tr_4(n) の最大値
 \end{aligned}$$

### 【0057】

#### 【数11】

$$\begin{aligned}
 sTe(n) &= (sTe_1(n), sTe_2(n), sTe_3(n), sTe_4(n)) \\
 &= 15 \times \left( \frac{Te_1(n) - Tr_{1_{min}}}{Tr_{1_{max}} - Tr_{1_{min}}}, \frac{Te_2(n) - Tr_{2_{min}}}{Tr_{2_{max}} - Tr_{2_{min}}}, \frac{Te_3(n) - Tr_{3_{min}}}{Tr_{3_{max}} - Tr_{3_{min}}}, \frac{Te_4(n) - Tr_{4_{min}}}{Tr_{4_{max}} - Tr_{4_{min}}} \right) \\
 (n &= 0, 1, \dots, 299) \\
 (Tr_{1_{min}}, Tr_{2_{min}}, Tr_{3_{min}}, Tr_{4_{min}}) &: Tr_1(n), Tr_2(n), Tr_3(n), Tr_4(n) の最小値 \\
 (Tr_{1_{max}}, Tr_{2_{max}}, Tr_{3_{max}}, Tr_{4_{max}}) &: Tr_1(n), Tr_2(n), Tr_3(n), Tr_4(n) の最大値
 \end{aligned}$$

### 【0058】

従来、このビット列への符号化では表1に示すバイナリ符号や、グレイ符号が用いられてきた。ところが、この符号化法を用いる場合、次に示す問題がある。つまり、筋電の特徴パターンは、図6に示した例のように、動作毎に任意の連続的な領域に分布していることが多い。

### 【0059】

そのため、この特徴パターンを論理回路によって識別するには、任意の連続した特徴量の値に対して1を出力する回路が必要である。しかしながら、グレイ符号の場合は、連続した値に対して1を出力するために、複雑な回路が必要な場合がある。以下に示す例では、符号化したビット列、つまり回路の入力信号を数12と表すこととする。

### 【0060】

#### 【数12】

$X_1 X_2 X_3 X_4$

### 【0061】

例えば、連続区間0～7に対してのみ1を出力する場合は、数13で表される回路によって、この区間の値に対して1を出力できる。

【0062】

【数13】

$$\overline{X_1}$$

【0063】

ところが、連続区間5～12の時だけ1を出力する場合は、3つの積項（入力信号の論理積）から成る回路（数14）が必要である。進化型チップによる回路合成では、一般的に、構成する入力信号の数が多い積項（例：4つの入力信号からなり、数12で表される積項や、上の例に示した積項（数15）や積項（数16）など）を多く必要とする回路の合成に時間がかかることが知られている。

【0064】

【数14】

$$X_2X_4 + X_2\overline{X_3}\overline{X_4} + X_1X_3\overline{X_4}$$

【0065】

【数15】

$$X_2\overline{X_3}\overline{X_4}$$

【0066】

【数16】

$$X_1X_3\overline{X_4}$$

【0067】

このため、グレイ符号を用いた場合、識別回路⑥の合成に時間がかかる場合があり、これが、識別精度を下げる一因になっていると考えられる。これに対して、冗長符号を用いて符号化を行うことにより、識別回路を単純な論理回路で実現できるため、進化型チップによる識別回路の合成を高速化でき、識別精度を改善できる。

【0068】

そこで、作成したトレーニングパターンを用いて進化型チップの回路を合成し、

トレーニングパターンによって識別精度の評価を行ったところ、冗長符号を用いることによって、識別精度を、13人の平均で3.1%、最大で10.5%改善できることを実証した。

#### 【0069】

さらに、回路合成に要した時間として、トレーニングパターンを用いて回路を評価した回数の、13人の平均値を比較すると、冗長符号を用いない場合に451582.71回必要であったのに対し、冗長符号を用いた場合には314333.91回であった。つまり、提案手法によって69.6%にまで短縮した。

#### 【0070】

本発明の特徴とする符号化装置⑧では、特徴パターンを表2に示す冗長符号を用いて、ビットパターンに符号化する。この符号はジョンソンカウンタの出力パターンとして用いられている符号であるが、冗長な符号化法であるため、4ビットの場合は0~7までしか符号化できない。

#### 【0071】

この冗長符号の特徴は、任意の連続的な値に対して、単純な回路だけで1を出力できることである。つまり、この符号では、2つの値の差が1である場合（区間長2）に、それらの符号は常に1ビットだけ異なり、差が2の場合は2ビット、差が3の場合は3ビットだけ異なるように設計されている。

#### 【0072】

例えば、1の冗長符号は0001、2の冗長符号は0011であるので、これらは1ビットしか異ならない。また、1と3の場合は、それらの符号が0001と0111であるので2ビットだけ異なる。このため、区間長2の任意の連続的な値に対して1を出力する場合は、3つの入力信号で構成される積項を持つ回路が必要である。例えば、区間1~2の場合は数17で表される積項を持つ回路によって、これらの区間にに対して1を出力可能である。

#### 【0073】

【数17】

$\overline{X_1 X_2} X_4$

【0074】

同様に、区間長3の場合は2つ、区間長4の場合は1つの入力信号で構成される積項だけで、それらの区間の値に対して1を出力する識別回路を実現できる。さらに、区間長が5～7の場合は、区間長4の場合に必要な積項を2つ組み合わせることによって実現できる。

【0075】

例えば、区間1～6（区間長6）の場合は区間1～4に対して1を出力する積項（数18）と、区間3～6に対して1を出力する積項（数19）の論理和（数20）によって、この区間の値に対して1を出力する回路を実現できる。

【0076】

【数18】

$X_4$

【0077】

【数19】

$X_2$

【0078】

【数20】

$X_2 + X_4$

【0079】

このように、本発明は、連続的な領域に分布する筋電パターンを識別する識別器を簡単な回路で実現できる。

また、本発明は、冗長符号化法を用いることで、単純な演算処理装置や、表引き装置によって実現できるため、小型化、低価格化が可能である。

これによって、さらに、制御可能な制御対象の数を増やすことが可能となり、このため、筋電をインターフェースとする装置の普及を促進することが可能となる。

What is claimed is:

1. 複数の筋肉の協調活動から発生する活動電位である筋電パターンを皮膚表面上で表面電極群により測定し、測定された筋電パターンから特徴パターンを抽出し、抽出された特徴パターンを冗長符号によってビットパターンに符号化し、符号化されたビットパターンのパターン識別を行って出力制御信号を生成する筋電パターンの識別方法。
2. 複数の筋肉の協調活動から発生する活動電位である筋電パターンを皮膚表面上で測定する表面電極群と、測定された筋電パターンから特徴パターンを抽出する特徴パターン抽出装置と、抽出された特徴パターンを冗長符号によってビットパターンに符号化する符号化装置と、符号化されたビットパターンの識別を行って出力制御信号を生成するパターン識別器から成る筋電パターン識別装置。

## 要 約 書

複数の筋肉の協調活動から発生する活動電位である筋電パターンを皮膚表面上で表面電極群により測定し、測定された筋電パターンから特徴パターンを抽出する。この抽出された特徴パターンを冗長符号によってビットパターンに符号化し、符号化されたビットパターンのパターン識別を行って出力制御信号を生成する。